

2017-8

## A Case Study for eCampus Spatial: Business Data Exploration

James Carswell

*Technological University Dublin, [jcarswell@tudublin.ie](mailto:jcarswell@tudublin.ie)*

Thanh Thao Pham Ti

*National University of Ireland, Maynooth*

Andrea Ballatore

*University of California, San Francisco*


Junjun Yin

*Technological University Dublin, [junjun.yin@tudublin.ie](mailto:junjun.yin@tudublin.ie)*

Linh Truong-Hong

*Technological University Dublin, [linh.truonghong@tudublin.ie](mailto:linh.truonghong@tudublin.ie)*

Follow this and additional works at: <https://arrow.tudublin.ie/aaschmedbk>

 Part of the [Databases and Information Systems Commons](#), and the [Graphics and Human Computer Interfaces Commons](#)

---

### Recommended Citation

Thi, Thanh Thoa Pham, Andrea Ballatore, Junjun Yin, Linh Truong-Hong and James D. Carswell. "A Case Study for eCampus Spatial: Business Data Exploration." *Handbook of Research on Geospatial Science and Technologies*. IGI Global, 2018. 240-270. doi:10.4018/978-1-5225-3440-2.ch016

This Book Chapter is brought to you for free and open access by the School of Media at ARROW@TU Dublin. It has been accepted for inclusion in Books/Book chapters by an authorized administrator of ARROW@TU Dublin. For more information, please contact [yvonne.desmond@tudublin.ie](mailto:yvonne.desmond@tudublin.ie), [arrow.admin@tudublin.ie](mailto:arrow.admin@tudublin.ie), [brian.widdis@tudublin.ie](mailto:brian.widdis@tudublin.ie).



This work is licensed under a [Creative Commons Attribution-NonCommercial-Share Alike 3.0 License](#)

# Handbook of Research on Geospatial Science and Technologies

Joyce Gosata Maphanyane  
*University of Botswana, Botswana*

Read Brown Mthanganyika Mapeo  
*University of Botswana, Botswana*

Modupe O. Akinola  
*University of Lagos, Nigeria*

A volume in the Advances in Geospatial  
Technologies (AGT) Book Series



Published in the United States of America by

IGI Global  
Engineering Science Reference (an imprint of IGI Global)  
701 E. Chocolate Avenue  
Hershey PA, USA 17033  
Tel: 717-533-8845  
Fax: 717-533-8661  
E-mail: [cust@igi-global.com](mailto:cust@igi-global.com)  
Web site: <http://www.igi-global.com>

Copyright © 2018 by IGI Global. All rights reserved. No part of this publication may be reproduced, stored or distributed in any form or by any means, electronic or mechanical, including photocopying, without written permission from the publisher. Product or company names used in this set are for identification purposes only. Inclusion of the names of the products or companies does not indicate a claim of ownership by IGI Global of the trademark or registered trademark.

Library of Congress Cataloging-in-Publication Data

Names: Maphanyane, Joyce Gosata. | Mapeo, Read Brown Mlthanganyika. | Akinola, Modupe O.

Title: Handbook of research on geospatial science and technologies / Joyce Gosata Maphanyane, Read Brown Mthanganyika Mapeo, and Modupe O. Akinola, Editors.

Description: Hershey PA : Engineering Science Reference, [2018]

Identifiers: LCCN 2017017588 | ISBN 9781522534402 (hardcover) | ISBN 9781522534419 (ebook)

Subjects: LCSH: Geographic information systems--Research. | Geospatial data. | Remote sensing.

Classification: LCC G70.212 .H348 2018 | DDC 621.36/78--dc23 LC record available at <https://lccn.loc.gov/2017017588>

This book is published in the IGI Global book series Advances in Geospatial Technologies (AGT) (ISSN: 2327-5715; eISSN: 2327-5723)

British Cataloguing in Publication Data

A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The views expressed in this book are those of the authors, but not necessarily of the publisher.

For electronic access to this publication, please contact: [eresources@igi-global.com](mailto:eresources@igi-global.com).

# Chapter 16

## A Case Study for eCampus Spatial: Business Data Exploration

**Thanh Thoa Pham Thi**

*National University of Maynooth, Ireland*

**Andrea Ballatore**

*University of California – Santa Barbara, USA*

**Junjun Yin**

*Dublin Institute of Technology, Ireland*

**Linh Truong-Hong**

*University College Dublin, Ireland*

**James D. Carswell**

*Dublin Institute of Technology, Ireland*

### ABSTRACT

*Location based querying is the core interaction paradigm between mobile citizens and the Internet of Things, so providing users with intelligent web-services that interact efficiently with web and wireless devices to recommend personalised services is a key goal. With today's popular Web Map Services, users can ask for general information at a specific location, but not detailed information such as related functionality or environments. This shortcoming comes from a lack of connection between non-spatial "business" data and spatial "map" data. This chapter presents a novel approach for location-based querying in web and wireless environments, in which non-spatial business data is dynamically connected to spatial base-map data to provide users with spatially-enabled attribute information at particular locations. The proposed approach is illustrated in a case study at the National University of Ireland in Maynooth (NUIM), where detailed 3D campus building models were constructed. Non-spatial university specific business data such as the functionalities and timetables of class rooms/buildings, campus news, noise levels, and navigation are then explored over the web and presented as both mobile and desktop web-services.*

DOI: 10.4018/978-1-5225-3440-2.ch016



## INTRODUCTION

Tools for spatial exploration are provided today by free products such as Google Maps and Google Earth with satellite and street views. Users can search a street address on the map, then explore the location in street view mode, or find how to reach a certain address or location providing options like pedestrian route or driving route, etc. The query tools available in these products are usually limited to keyword-based search. At larger local scales, where detailed 3D geometries and associated business data are needed, there is a general lack of related information and search functionality for in-depth exploration of an area (PhamThi, Truong-Hong, Yin, and Carswell, 2013). Moreover, such generic products do not provide domain-specific information, highly relevant for the users.

For instance, the following types of questions cannot be answered when interacting with Google Maps/Earth on a typical university campus: what classes are scheduled in that room over there? Whose office window is that up there? What buildings/classrooms/labs can I actually see around me while standing at a specific location on campus? In order to answer these types of specific questions, Location Based Services (LBS) need to link spatial (map) data with non-spatial business data. Spatial data deals with detailed topology and geometry or coordinates of objects while business data can describe the attributes or semantic aspect of a related object in some business domain, e.g. a university campus (PhamThi et al., 2013)).

In general, geospatial data tend to change at a slower pace than business data. Furthermore, conventional business data is often managed and produced by traditional enterprise information systems, often ignoring the spatial dimension. Thus linking spatial data and business data in one application helps to enrich the user experience by fulfilling more specific user needs. In particular for task specific decision making applications that need access to detailed local scale data typically found in zoos, museums, hospitals, shopping malls, a university campus or retail/office park settings.

Business data is often indirectly or virtually associated to spatial data via its location given by a generic address, a room number, a building name, or even lat/long coordinators of the business location. Such business data can provide further detailed information to users about the objects in question and be tailored or application domain dependent. For instance, in a university campus environment, the business data involved may be in the form of class schedules for a specific room, lists of equipment installed in a lab, office hours or contact details for a lecturer, today's special meal deal in a cafeteria, etc.

Typically, 2D "footprint" data provides just flat geometry representation of physical objects (e.g. buildings) in the horizontal plane. But for this linked spatial/non-spatial data application, we need spatial and non-spatial attribute details in the vertical dimension as well for in-depth 3D BIM (building information modelling) data query operations. Depending on how much 3D geometry data is captured and how available any related business data is, more informative task specific answers can be provided to the user. For instance, 3D BIM data of a building can include detailed digital representations of physical and functional characteristics for its different floors, rooms, windows, and doors where all "things" are potentially available for interrogation.

As a case study, we describe in detail the prototype of a system in which these ideas were implemented. Within the framework of the StratAG project (<http://stratag.ie/>), an eCampus Demonstrator was developed for the National University of Ireland Maynooth (NUIM), in collaboration with Dublin Institute of Technology (DIT), and University College Dublin (UCD). This web-based GIS application aims to assist users in exploring and analyzing their surroundings within a detailed data environment; in our case, domain specific university business data is linked together with 3D spatial built environment

data to answer more task specific user queries. The application addresses two types of users: public users (e.g. visitors) and local users (e.g. students and staff). Access privileges and query levels depend on user type. For example, visitors are presented with a campus map for general information querying of buildings and rooms. Buildings and room information include building images, opening times, class schedules, room facilities. Visitors are also provided general campus news, events and utilities for navigation to various buildings or rooms or any other locations on campus using different routing options. In addition to these functionalities, staff and students are able to visualize their individual class schedules together with personalised news feed and events tailored to their academic and social interests.

Each project partner in the StratAG cluster is responsible for developing different functionality in the eCampus Demonstrator such as utilities for 2D/3D directional querying (DIT), for path navigation assistance (NUIM), for personalising news and events according to user interests (UCD), or for detailed 3D modelling of the campus infrastructure itself (NUIM/DIT). RESTful web-services (OGC, 2007) were chosen as the deployment technology of these distributed components thanks to its simplicity when applied particularly to the geospatial domain (Mazzetti, Nativi and Caron, 2009), (Guinard, Trifa and wilde, 2010), (Kurtagic, Birch and Zeiss, 2009). Regarding 3D maps, there are several commercial and free mapping products available that allow users to incorporate 3D building models for visualization and interaction applications. Of these, Google Earth (GE) was selected for displaying the 3D buildings used in this work because it is free and widely popular with both web and wireless GIS users.

In this chapter, we first introduce our approach to 3D BIM with a detailed workflow and description of each step in the modelling process. The 3D building information is then imported into a spatial database as well converted to a GE readable format for 3D map display. Then we present the system architecture of the eCampus Demonstrator based on a Resource Oriented Architecture (ROA) model. The main functionalities of eCampus are described in detail together with its graphical user interfaces. Finally we draw conclusions and give some possible direction for future work.

## **RELATED ECAMPUS APPLICATIONS**

Assisting people with exploring an area, such as a university campus, with a mapping interface is very useful, and is not a new idea in itself. Many existing projects provide this type of service. Salient examples include:

Kent State University campus maps (Kent State University, 2013) is a web-based application. It provides an interactive 2D map with detailed information and images of each building in the campus. It also highlights specific locations on the campus such as computer labs, parking, sculpture walk, and residence halls. The *Get* direction functionality allows to find pedestrian/driving/cycling routes between two locations, e.g. using building names. However, this application provides very basic query/search functionality overall, has no 3D maps, or any form of personalisation.

The Berkeley University of California Interactive Map (Berkley University, 2013) provides an isometric campus map with clickable buildings to provide detailed information and an image of each building. There is also a list of building names placed outside the map window from where users can choose a name, then the corresponding building object on the map is highlighted. This application has no search facility, no routing service, and provides limited spatial support.

The University College Dublin Mobile services (University College Dublin, 2013), released natively on Android and iPhone, has the following functionality: Campus maps, details about places, and tours;

Campus directory; Access to Library; Access to e-learning facility (Blackboard); Schedule of general events (lectures, concerts, etc.); Campus news; Image search on university archives; and Emergency numbers. The application has two main versions, one for staff and one for students, with different permissions. Although this application provides many useful services, particularly in relation to campus events, its spatial support is still very limited. It has only a simple, non-interactive 2D map, with no routing functionality.

The University of Karabuk, Turkey (Kaharaman, Karas and Rahman, 2011) is a web-page application allowing users to explore the university campus in 3D. It provides information at the building level and points of interests, but room details have not been fully developed, and the implementation does not provide utilities to further query the area beyond its physical, spatial nature.

It is clear from current related work that functionality for exploring non-spatial business data of sub-objects like rooms, windows, and doors to retrieve the purpose, content, or schedule of a specific room in a building is still an evident limitation of existing detailed data location-based services (LBS) type applications. This is largely due to insufficient geometric detail of online building models available on today's web mapping platforms, and subsequently a general lack of spatially linked business data.

## **PREPARING 3D MODELS FOR GOOGLE EARTH INTEGRATION**

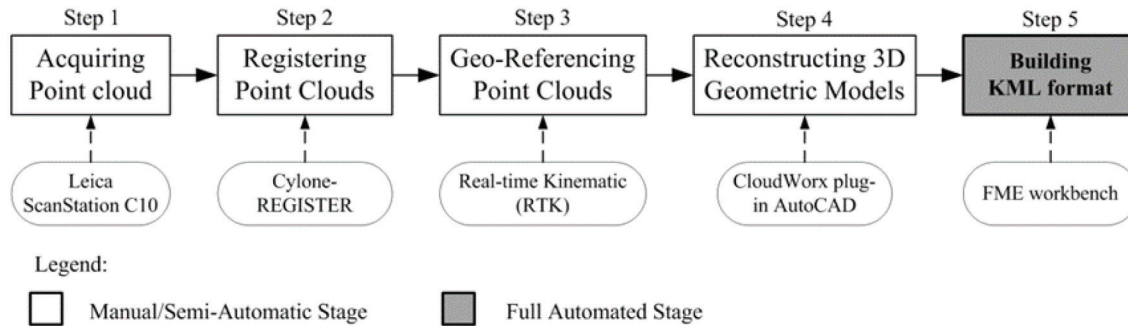
The raw data used to reconstruct detailed 3D building models for 3D city maps can be obtained from various sources through a wide range of techniques. With recent developments in photogrammetry and remote sensing, building models can be automatically reconstructed given the geometric resolution of satellite imagery and Light Detection and Ranging (LiDAR) point cloud data (Haala and Kada, 2010). The geometric and semantic properties of 3D models are typically stored in five consecutive levels-of-detail (LoD), in which LoD0 defines a coarse regional scale model while LoD4 denotes architectural building models with detailed walls, roof structures, balconies, interior structures, and detailed vegetation and transportation objects. An advantage of the LoD approach is the coherent modelling of semantics and geometric/topological properties together at each level, where geometric objects get assigned to semantic objects. In order to obtain our objective, the 3D building modelling has to be at least at LoD3 level.

To accomplish this, a workflow was developed for reconstructing building models from LiDAR point clouds suitable for real-time display, and then loaded these models into GE and a spatial database (Truong-Hong, Pham Thi, Yin, Carswell, 2013).

### **Workflow for eCampus 3D Modelling**

In this work, LiDAR point clouds of campus buildings was first acquired using Leica ScanStation C10, which was controlled by proprietary software Cyclone installed on a laptop linked to the scanner (Leica Geosystems, 2011a). Subsequently, the point clouds were registered and geo-referenced within the Cyclone environment and the 3D building models manually created using AutoDesk and a CloudWorx plug-in. CloudWorx offers many manipulation and editing tools to assist users to trace or auto fit lines, arcs and polylines to 3D point cloud data (Leica Geosystem, 2011b). Finally, by employing FEM Workbench (FME, 2013), the 3D building model's underlying CAD geometry is transformed to KML format to allow for display by web-based mapping applications in Google Earth.

Figure 1. The workflow of 3D BI modelling



This modelling methodology was applied to the north campus of the NUIM study area. This was selected because the area contains a mixture of simple and complex buildings with various architectural styles (e.g. historic and modern buildings), which can be most problematic when reconstructing 3D building models. The tallest building is 20 m.

## Five Steps for Generating GE Building Models for Real-Time Online Display

### Step 1: Acquiring Point Cloud

In step 1, 14 buildings in the NUIM study area were scanned using a Leica ScanStation C10. Due to the complexity of some buildings, multiple scan stations were set up around a building to ensure sufficient point cloud coverage (Table 1). Each dataset was stored in a local coordinate system centred around the

Table 1. Configuration of eCampus Datasets

No.	Building name	No. scan stations	No. points (million)
1	Arts block	8	16.19
2	John Hume	16	5.77
3	Science	12	13.49
4	Callan	18	20.31
5	Engineering	15	15.45
6	Canteen and sport center	9	15.38
7	Hamilton and Rye Hall	36	44.87
8	Student Union	6	7.41
9	Iontas	12	6.00
10	Education House	18	18.86
11	St. Anne's	12	14.55
12	St. Catherine's	11	6.60
13	Student Services	13	13.32
14	Auxilia	17	8.09

## ***A Case Study for eCampus Spatial***

location of the scanner. A resolution of 10 mm point spacing was set for the data collection process. In fact, as north campus buildings are sparsely distributed, this allowed the terrestrial laser scanning (TLS) to acquire sufficient point clouds of some roofs as well.

### **Step 2: Registering Point Cloud**

To prepare the point clouds for building reconstruction, the datasets from each scan station were registered and merged into one large dataset stored in a single coordinate system using the Cylone-register module. In this process (step 2), a source dataset is manually merged to a target dataset by mapping pairs of reference points from the target and source datasets. There are three reference points for each dataset, and are often selected from sharp angular features on the building facade, for example window corners or lintel edges. The registration process is completed when the average distance between any two data points in an overlapping region is less than a threshold, for example, Truong-Hong (2011) uses a threshold of 5mm.

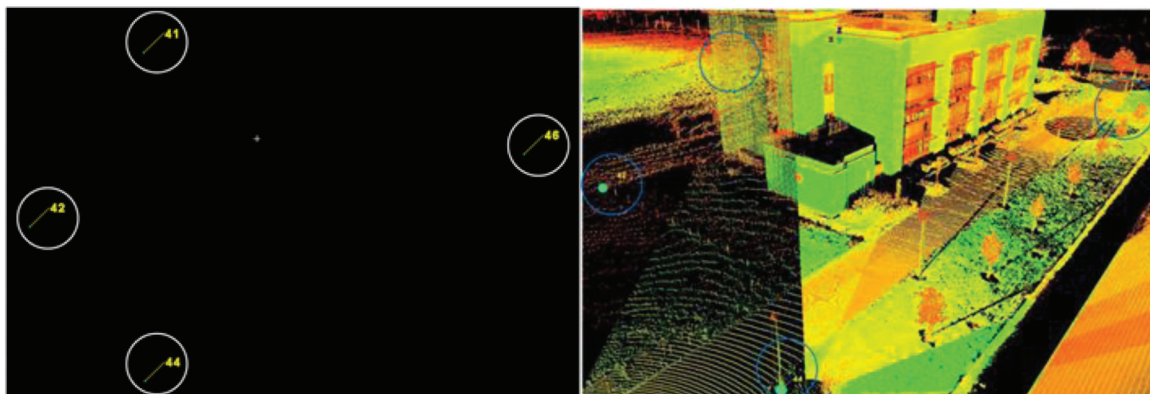
### **Step 3: Geo-Referencing**

The Irish National Grid coordinate system was assigned to the point cloud in the final dataset to geo-reference the entire campus model and allow for geographically accurate visualizations (step 3). Real Time Kinematic (RTK) positioning was used to get accurate X, Y, and Z ground control points. Approximately, 50 observations were taken in total on the North campus with a maximum elevation error less than 1cm. To obtain control points at each chosen location, residence times varied from 2-4 minutes for each control point. After importing RTK ground control points into the Cylone program, common data points from the LiDAR building dataset were mapped to the RTK ground control points within the ScanWorld module of the Cylone program (Figure 2).

### **Step 4: Reconstructing 3D Models in AutoCAD**

After registering and geo-referencing the building point clouds, they were imported into the AutoCAD environment through a Leica plug-in called CloudWorx, which is a tool for viewing and working with

*Figure 2. Process of geo-referencing point clouds*



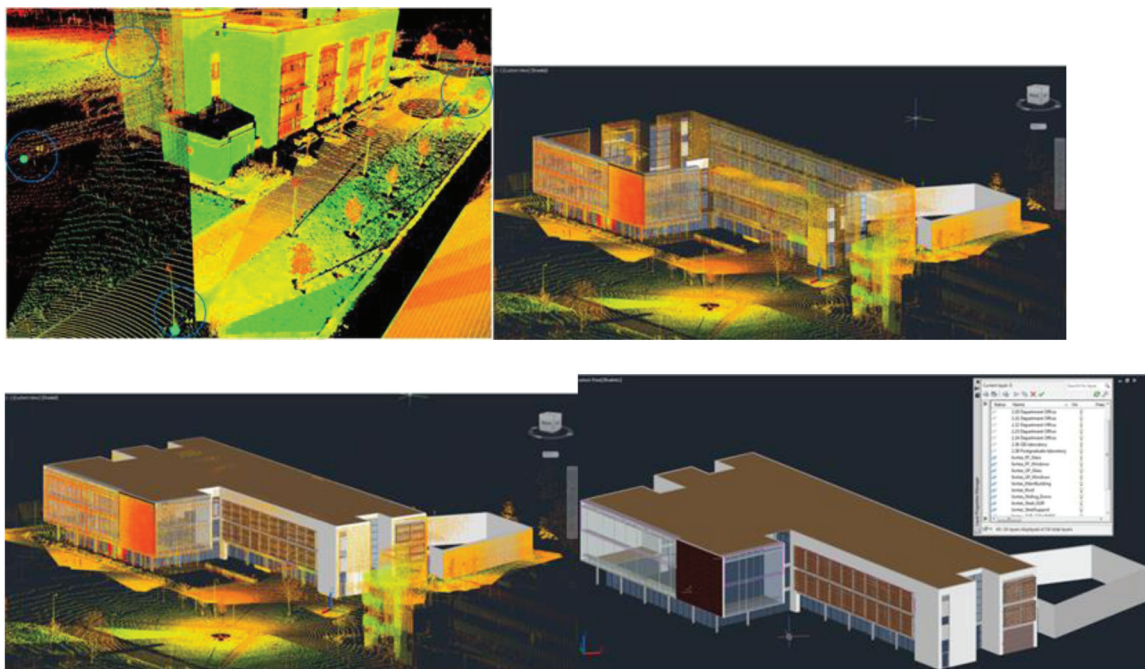


slices of point cloud data to reconstruct 3D solid models directly from LiDAR point clouds (Leica, 2011b). For example, Figure 3 illustrates the process of building reconstruction (Step 4). During reconstruction, details of the building, e.g. windows/door frames or balconies and photographs of the building are used as reference.

The solid building model created in the previous step has limitations concerning access to its object and sub-object attributes (such as building name or room name) in Google Earth. A restriction of GE when displaying/querying 3D solids is that linked attribute information cannot be accessed by clicking (pointing/selecting) anywhere on the solid shape. To overcome this, the solid building shape must be converted to 3D polygon data. An overlay layer of 3D polylines/polygons around each window/door border in the CAD drawing was therefore added as GE does recognize clicking anywhere inside a polygon shape to select/query the object.

Both 3D solids and 3D polylines are managed in separate layers in the AutoCAD database. There are two groups of layers: (i) stored solid components involving exterior vertical walls, window/door frames, roofs, and balconies and (ii) stored polygons of window/door extents. The layer name is assigned based on its group. For example, with the Iontas building model in Figure 3, if the layer belongs to group (i), its name incorporates the building name plus the component name (e.g. a main wall is stored in the layer named “Iontas\_MainWall” while a window frame is stored in the layer named “Iontas\_WindowFrames”). Similarly, if the layer is in group (ii), the layer name describes the number and name of the room. For example, in the Iontas building, the window polyline of room 2.36 GIS Laboratory is stored in the layer named “2.36 GIS Laboratory”. This data management approach allows for efficient geometry extraction of building components in the next step.

Figure 3. Process of Iontas building modelling from LiDAR point cloud data



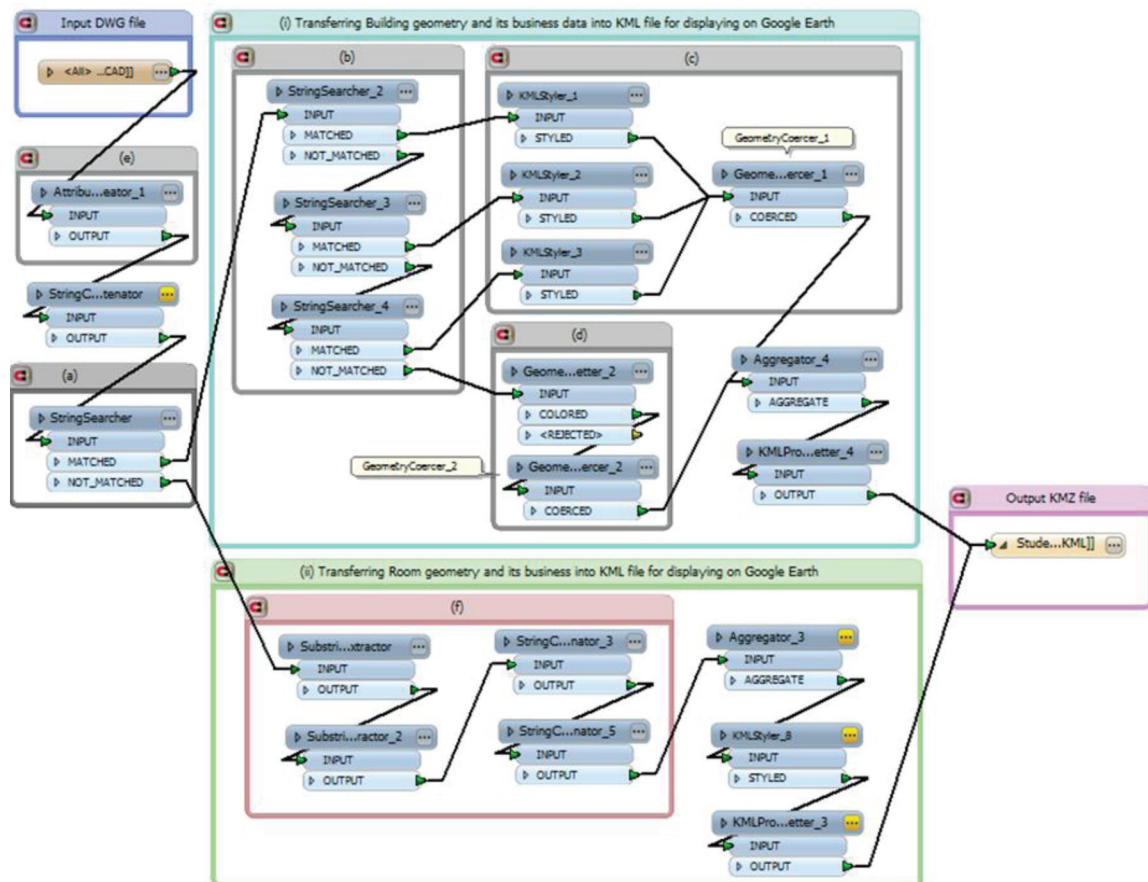
In summary, 3D solids and 3D polygons together represent the campus building models and their associated rooms linked to any available metadata information.

### Step 5: Transforming 3D Building Models to KML Format for Google Earth Display

GE allows users to upload 3D building models for direct visualization in the GE environment or for inclusion in a webpage using their application programming interface (API). When retrieving data for visualization, a client (desktop or mobile) queries for data either from temporary cache memory or from GE databases. In order to include 3D building models in GE (step 5 in the workflow), the original models stored in AutoCAD format (DWG) is transformed into Google Earth (KML) format using the FME Workbench utility (FME, 2013).

The KML format supports both solid and polygon data types where 3D solids are stored in the COLLADA interchange file format. A workflow was developed to transform building model CAD files to KML through two schemas: (i) for geometries of the building and (ii) for geometries representing rooms (Figure 4).

*Figure 4. A workflow to transform 3D building models from DWG to KML*



The geometries of building components (e.g. main walls, roofs, balconies window/doors) are managed by two groups of layers in an AutoCAD database. A “StringSearching” transformer [box (a) in Figure 4] was used to separate geometries of the building and rooms based on the AutoCAD layer names. Subsequently, a series of “StringSearching” transformers [box (b) in Figure 4] continuously extracted geometries of each component of the building model. 3D solids of the building components in a CAD file are now transferred into KML file as polygons [box(c) in Figure 4] and solids [box (d) in Figure 4] using the “GeometryCoercer” transformer. Of which, 3D solids are only used for geometries of window/door frames while polygons are applied to all other objects. This step allows us to assign an appearance to each building component by filling with either a colour or texture. In this study, the appearance of a building is defined by a filling colour through the “KMLStyle” transformer [box(c) in Figure 4] for polygons and “GeometryCoercer” transformer [box(d) in Figure 4] for solids. Additionally, the geometries of polyline rooms automatically get transformed to 3D polygons in the KML file.

## **ECAMPUS ARCHITECTURE**

The eCampus Demonstrator is a browser-based application based on 2D OpenStreetMap (OSM) data and 3D Google Earth (GE) data is accessible to both desktop and mobile devices. It aims to help users explore in more detail the National University of Maynooth’s campus by providing them detailed maps and utilities for both 2D and 3D querying and visualisation. Different query functionality is provided so that users can ask questions by interacting with the map itself. For instance, they can ask: “What is that building over there?” by pointing at it with their mobile device; “What is the class schedule of this classroom?” by clicking on its window or by choosing a room name from a list; “What can I actually see around me?” when standing in a particular location on campus; they can also ask to visualise a route together with “directional images” (i.e. containing superimposed arrows pointing the way) between two buildings/rooms or any location by choosing a building/room from a list or by clicking on the map. Query results are visualised on the 2D/3D map afterwards overlaid with further business data where available.

The application architecture includes 3 layers: interface layer, web-services, and database layer (Figure 5).

### **Database Layer**

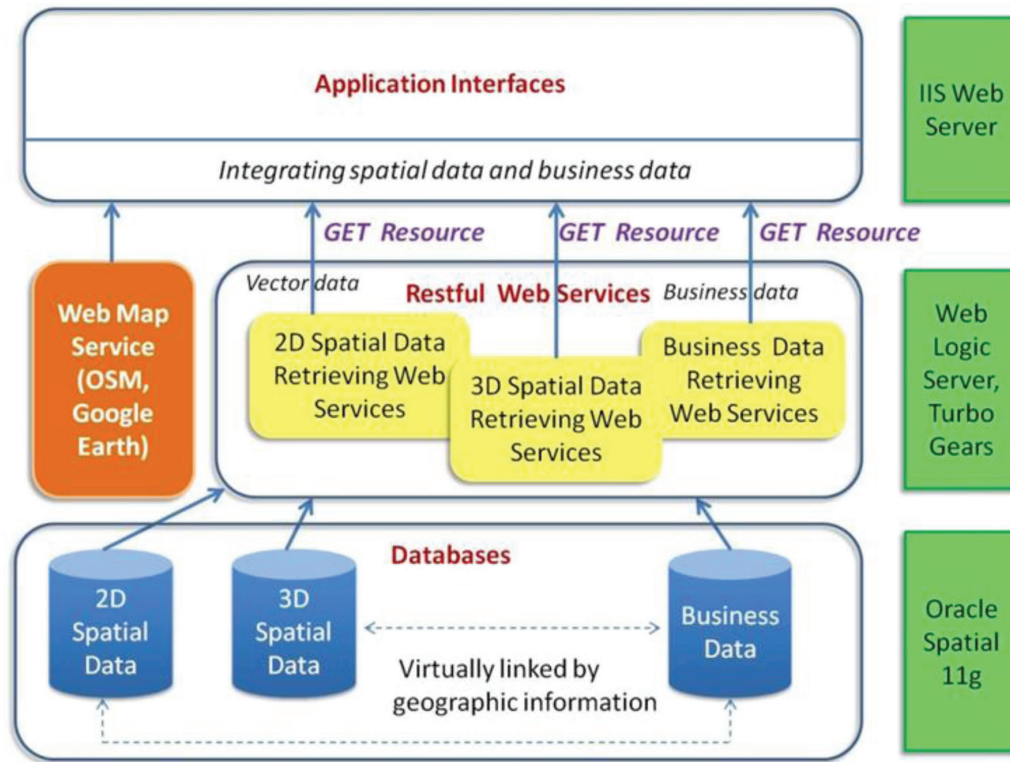
The 2D footprint and detailed 3D models of NUIM campus are hosted in Oracle Spatial 11g databases that serve the spatial data retrieving web-services. Other data related to indoor pathways, outdoor roads and building images are hosted in PostGIS databases. In the scope of this chapter, we describe in detail the development of 2D footprint and 3D building data.

### **2D Database Preparation**

Many online maps such as Google Maps and Bing Maps still have considerably poor data coverage over many areas, especially in those less populated, like Maynooth Ireland. As NUIM’s campus is located in such a rural area, OpenStreetMap (OSM) shows the advantage of collective efforts for Volunteered Geographic Information (VGI), where anonymous users contribute to OSM by uploading geographic features such as buildings, streets, point-of-interests (PoIs), etc. to complete the map coverage. In par-



Figure 5. Three Tier eCampus Demonstrator System Architecture



ticular, students and others from NUIM have already done this, which proved very beneficial in our case (Musliman, Abdul-Rahman, Coors, 2010). Therefore, many 2D campus footprints were first downloaded from OSM and then uploaded to our local Oracle Spatial DBMS where geometry data is stored in a single column data type of *SDO\_GEOMETRY* to define the geometry type (e.g. points, lines, polygons, solids, etc.), the dimension, and an array of x, y (and z for 3D) coordinates comprising points or vertices of campus objects.

### 3D Database Preparation

In Section 3, we presented the workflow for 3D BI modelling and export to KML format for displaying in GE. In fact, this 3D building information is also needed and stored in Oracle Spatial for visibility querying. After reaching Step 4 (3D BIM in AutoCAD), the Feature Manipulation Engine (FME) workbench utility was used to export the 3D objects to the spatial database. It is a similar process as when exporting to KML. This means from the result of Step 4, there are two groups of layers stored in the AutoCAD database: (i) stored solid components involving exterior vertical walls, window/door frames, roofs, and balconies and (ii) stored polygons of window/door extents. These layers are then exported to Oracle Spatial database. As mentioned in Step 4 above, the reason for this type of data management is because of “clicking” restrictions in Google Earth regarding pointing/selecting solid building objects and sub-object attributes directly from the display.

The process for exporting data from an AutoCAD DWG file into Oracle Spatial was developed and is shown in Figure 6. The upper box in Figure 6 depicts the workflow to transfer the external objects (exterior walls, roofs, balconies, doors and window frames) of the buildings into the table NUIMBUILDING. These objects were stored in a row of that table, with a datatype of *multisurface* (*SDO\_GTYPE* = 3007). The lower box illustrates the workflow to transfer the polygons of peripheries of all windows of each room to a row in the NUIMBUILDINGOBJECT table as datatype *heterogeneous* (*SDO\_GTYPE* = 3004).

Note that at this phase of the project, objects inside buildings are not modelled and captured. A room is represented by its windows, and selecting a room corresponds to selecting one of its exterior windows (or doors). Therefore, our spatial database contains mainly building objects and associated window/door objects.

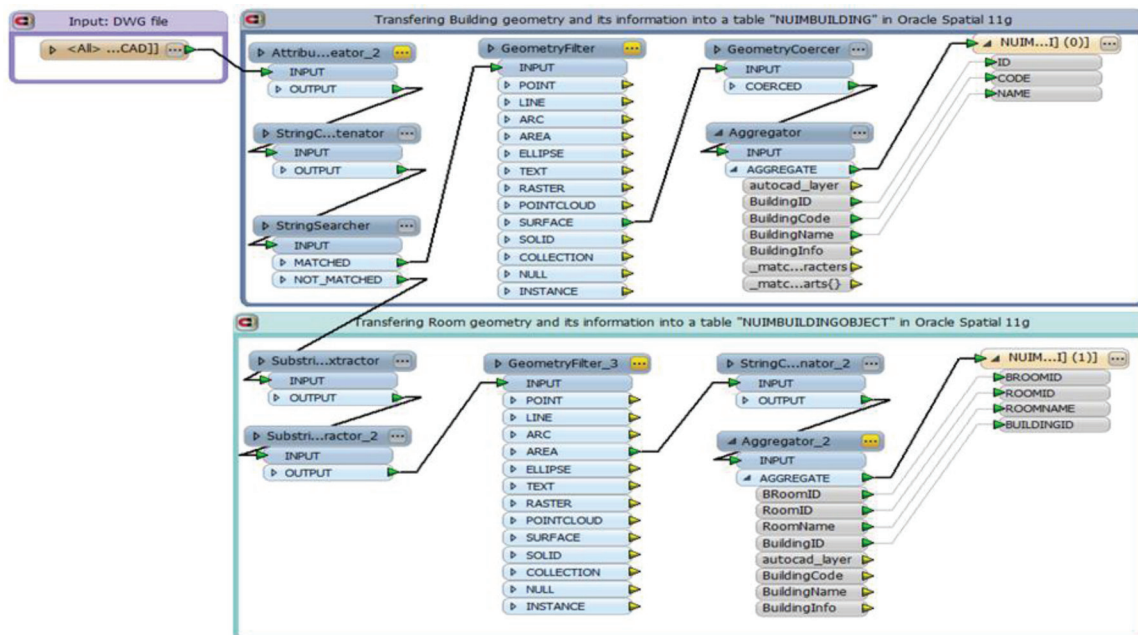
## Web-Services Layer

At the logical level, the spatial data retrieving web-services and business data retrieving web-services are installed in this layer.

## Spatial Data Retrieval Web-Services

Spatial data retrieval web-services include routing navigation, image retrieval for directional visualisation of routes, and 2D and 3D visibility based directional querying: 2D Isovist, point-to-select, and field-of-view, plus 3D Isovist, point-to-select, and frustum.

Figure 6. Process of transferring geometries of buildings and associated windows/doors together with their metadata information into Oracle Spatial 11g



## A Case Study for eCampus Spatial

These web-services are developed by different project partners on different platforms, but have the same deployment methods in the form of *RESTful* web-services. More specifically, an IIS (Internet Information Services) server is appointed to host the web-services. Query requests are constructed using standard HTTP calls containing a valid URL filled with the required query parameters.

The routing navigation web-service needs query parameters such as transport mode (e.g. pedestrian, driving, wheelchair, or directional images), and a starting and destination location in terms of longitude and latitude coordinates. The navigation web-service then returns a list of points (i.e. OSM object vertices along the route) in KML format. The eCampus applications then uses OpenLayers API to read this list and connect the KML points to display the route on the map for users. Users can also provide a building or room name for the start/end location. The corresponding coordinates of the building/room is retrieved from 2D and 3D spatial databases and passed to the web-service in this case. The following URL command describes a call to the pedestrian navigation web-service.

<http://147.252.87.12/RoutingWSEnhanced/pedtxt.ashx?longorigin=-6.597721790929617&latorigin=53.38485859013911&longdestination=-6.603740667958208&latdestination=53.38384112506816>

Example of the web-service route point list result in KML format:

```
<?xml version='1.0' encoding='UTF-8'?> <kmlxmlns='http://earth.google.com/
kml/2.0'>
<Document> <Placemark><description>479.5 Meter(s);5 Mins 27 S </description>
<LineString> <coordinates>
-6.59772179092962,53.3848585901391
-6.597728695735075,53.384872060169414
-6.5977571,53.3848575
...
-6.60374066795821,53.3838411250682</coordinates>
</LineString></Placemark> </Document></kml>
```

The web service result is displayed on the map as illustrated in Figure 7.

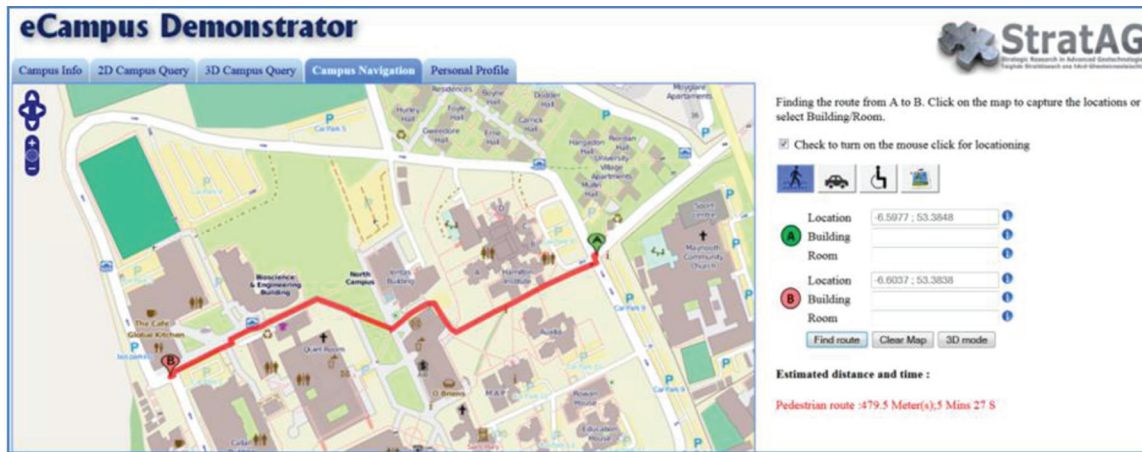
The directional images web-service provides a list of images (thumbnails) along the route augmented with superimposed arrows pointing the way. The parameters for this web-service are the same as for normal routing. The web-service returns in JSON format a list of image URLs, their location, and the view angle of each image wrt to the direction needed to follow. An example URL calling the directional image web-service follows:

[http://ggalessrv.cs.nuim.ie/eCampus/public\\_scripts/retrieveImage.php?startcamlat=53.384251080788125&startcamlon=-6.599744305677534&endcamlat=53.384283076607105&endcamlon=-6.602823481626215&raw](http://ggalessrv.cs.nuim.ie/eCampus/public_scripts/retrieveImage.php?startcamlat=53.384251080788125&startcamlon=-6.599744305677534&endcamlat=53.384283076607105&endcamlon=-6.602823481626215&raw)

Example of the web-service result in JSON format:

```
{ "img": [{ "lon": "-6.6004592", "lat": "53.3843123", "url": "http://ggalessrv.
cs.nuim.ie/eCampus/img/553387507.jpg", "angle": "341" }, { "lon": "-6.6006028", "la
```

Figure 7. The routing web-service result displayed on the map

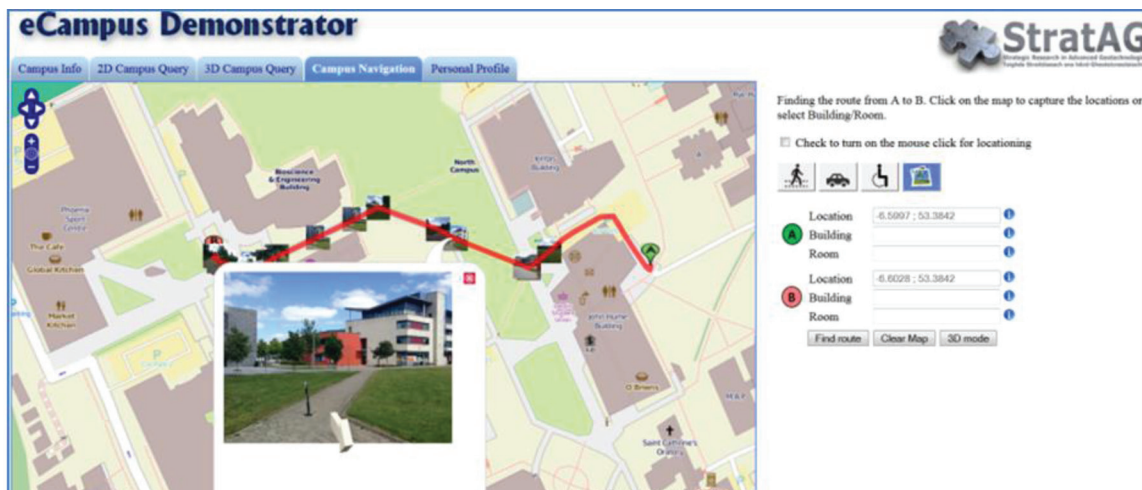


```
t":"53.384243","url":"http://ggalessrv.cs.nuim.ie/eCampus/img/553385583.jpg",
"angle":"210"},...
{"lon":"-6.602814734788404","lat":"53.38429296117096", "url":"http://ggalessrv.cs.nuim.ie/eCampus/img/553381119.jpg",
"angle":"347"}], "n":16}
```

Figure 8 describes how the web service result looks like on the map.

2D and 3D visibility based directional query web-services correspond to different viewing options. In relation to the different spatial data types in the database, these web-services are divided into two sub-groups, one applied to 2D OSM building footprints and the other applied to 3D GE building models. 2D Isovist view, 2D Field-of-view, 2D/3D Line-of-Sight (Point-to-Select), 3D Frustum and 3D Threatdome

Figure 8. The routing web-service result





## A Case Study for eCampus Spatial

are the different types of 2D and 3D spatial queries available, where different spatial search algorithms and web-services are developed for each (Carswell, 2010), (Carswell, Gardiner and Yin, 2010).

Below is an example URL constructed for requesting the 2D IsoVist web-service:

```
http://147.252.87.12/tdq/app/isovistView3/isovist3?tg_format=json&lat=53.384737414643&lng=-6.5997979498564&dist=95.5229132101768
```

This web-service returns all the buildings (and other database objects) that users can see around them according to their location and the view shed area around them.

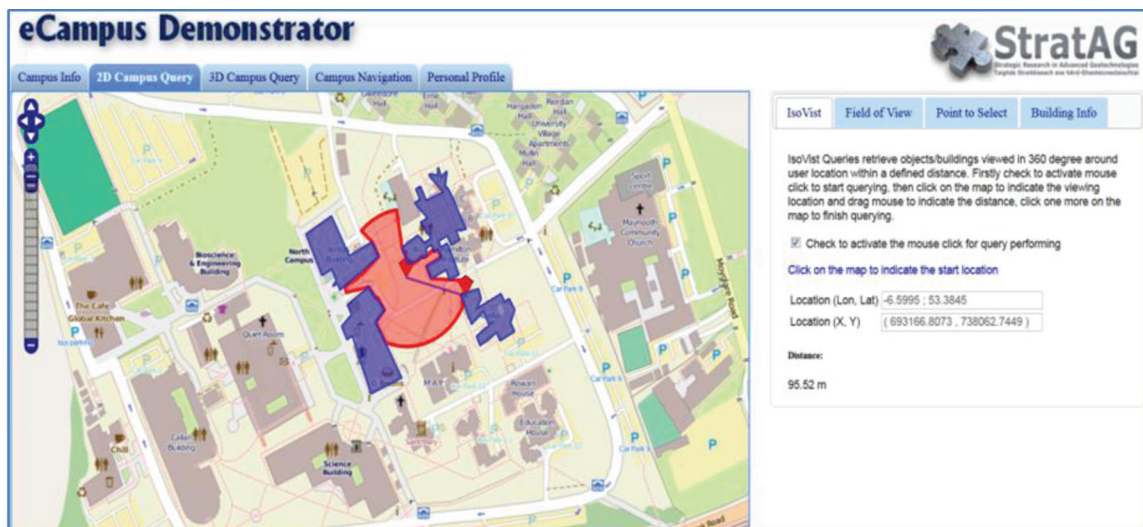
```
{"result": {"items": [{"Rye Hall", -6.5991027899, 53.3848764772}, {"Creche", -6.5995678094, 53.3850978962}, {"Tontas Building", -6.600487006000001, 53.3846852594}, {"John Hume Building", -6.6000432139, 53.3840204364}, {"Auxilia Building", -6.5983584111999996, 53.384220828000004}], "shape": [[-6.5997979498564, 53.38559980033004], ..., [-6.59992397015067, 53.38559651862579], [-6.5997979498564, 53.38559980033004]]}}
```

Based on the building name(s), another web-service is called to get the building shape (polygon) to draw the building boundaries on the map (Figure 9):

```
http://147.252.87.12/tdq/app/getSimplify/getSimplify?tg_format=json&name=Rye%20Hall&numberSelected=1
```

```
{"result": [[-6.598908099999996, 53.385156500000009, ..., -6.598908099999996, 53.385156500000009]]}
```

Figure 9. The Isovist web service result



## The Business Data Retrieval Web-Services

Useful business data (attributes) for student/staff users mainly relates to classroom schedules and facilities. *RESTful* web-services were developed to retrieve this information based on the room number of a building and querying a database containing NUIM business data. The web-service returns the class schedule in the form of a link to an html page.

Example URL calling a web-service to retrieve the timetable of a room

<http://147.252.87.12/ucd/retrieveTimetable.php?roomid=T9&semester=1>

The web-service result and how it looks on the map (Figure 10)

[http://147.252.87.12/eCampdemo/Timetable/T9\\_sem1.htm](http://147.252.87.12/eCampdemo/Timetable/T9_sem1.htm)

## Interface Layer

The interface layer consists of html pages displayed to users (client side) in a standard web browser. At this level, spatial data returned from 2D queries is visualised on 2D OSM maps as additional layers using OpenLayers API. Meanwhile for 3D queries, the results are returned in JSON format and drawn as Placemarks added to a Google Earth view. The integration of spatial data and business data is performed at the client side when attribute information about buildings and associated rooms/objects is found.

## ECAMPUS FUNCTIONALITY

The eCampus Demonstrator is developed for both web and wireless devices. However, there are some differences in the look and feel of the interfaces and user interactions between the desktop version and

Figure 10. The result of timetable retrieving web service is displayed on the 3D map



## A Case Study for eCampus Spatial

the mobile version. This is mainly due to limitations of the Google Earth API for mobile devices. In addition, on mobile devices information like user location, tilt, and compass readings can be captured automatically, while users need to indicate them when they use the desktop version. The application functionality and their organisation is depicted in Figure 11.

## Campus Exploration

This function displays in the main page of the application (Figure 12). It shows a 2D OSM map of the campus. Users can selectively click on each building (polygon) on the map to explore its detailed information.

Figure 11. eCampus functionality. There are 5 main functionalities in the eCampus Demonstrator. Each function provides a further subset of utilities (dashed boxes)

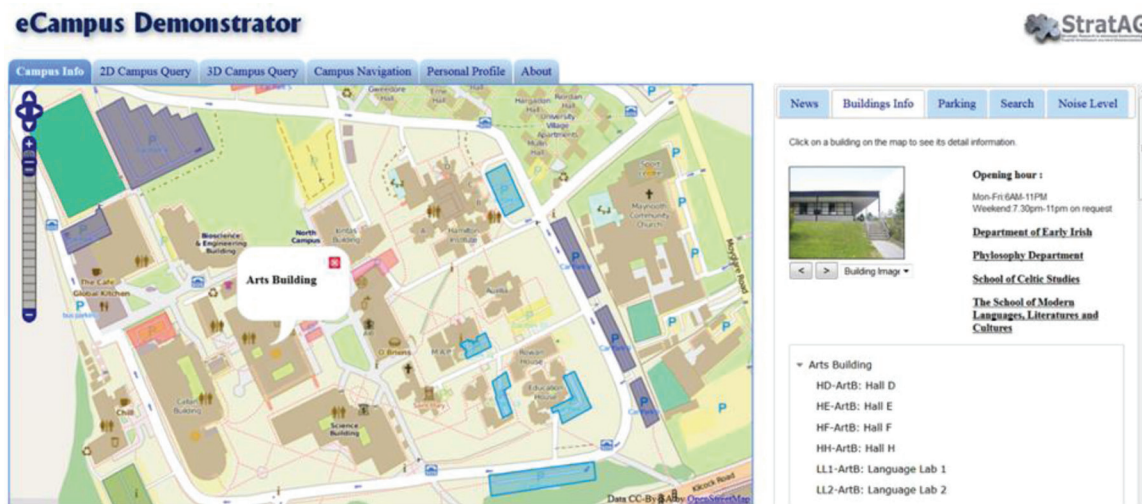
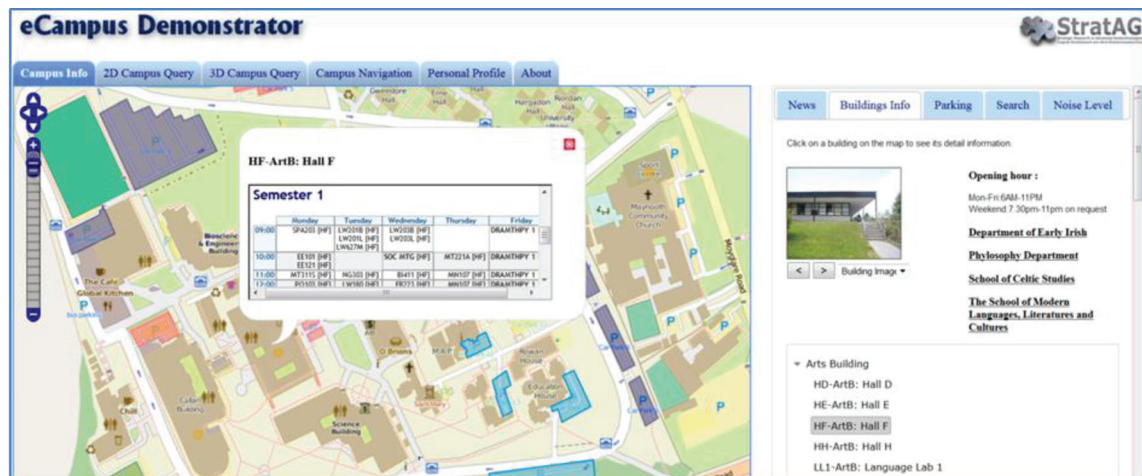


Figure 12. Building information





The available business data includes information on rooms within the building, faculties in each building, opening hours of the building, building images and architectural plans. Users can select rooms from the list to explore their detailed information such as opening hours, class schedules, facilities (Figures 12, 13). Parking information in which parking zones are indicated by colour code is also provided.

The application also provides a search utility where users enter a staff name (whole or part), the application provides an auto complete help to retrieve the staff office (with room and building name) (Figure 14).

The mobile interfaces for exploring the campus are depicted in Figure 15.

Figure 13. Room information

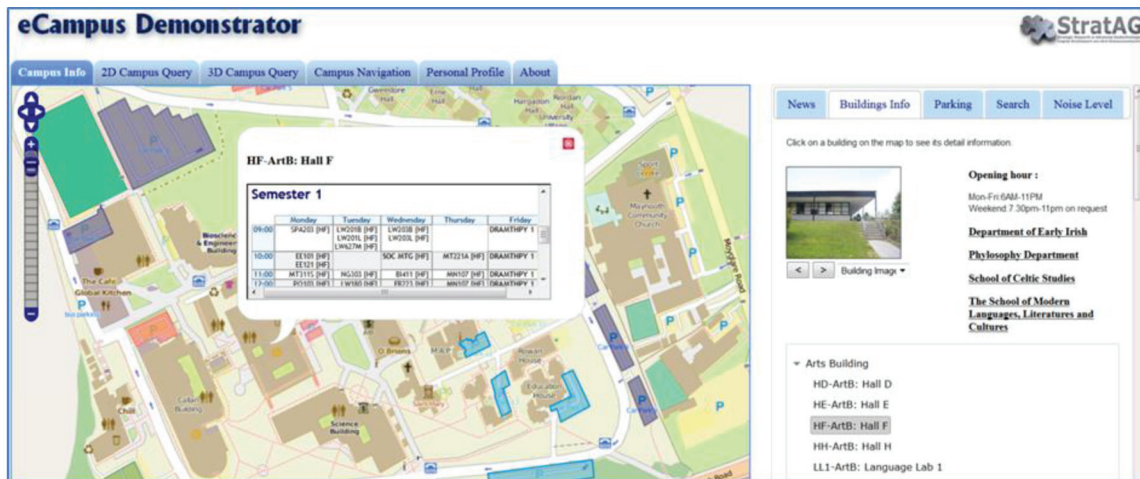
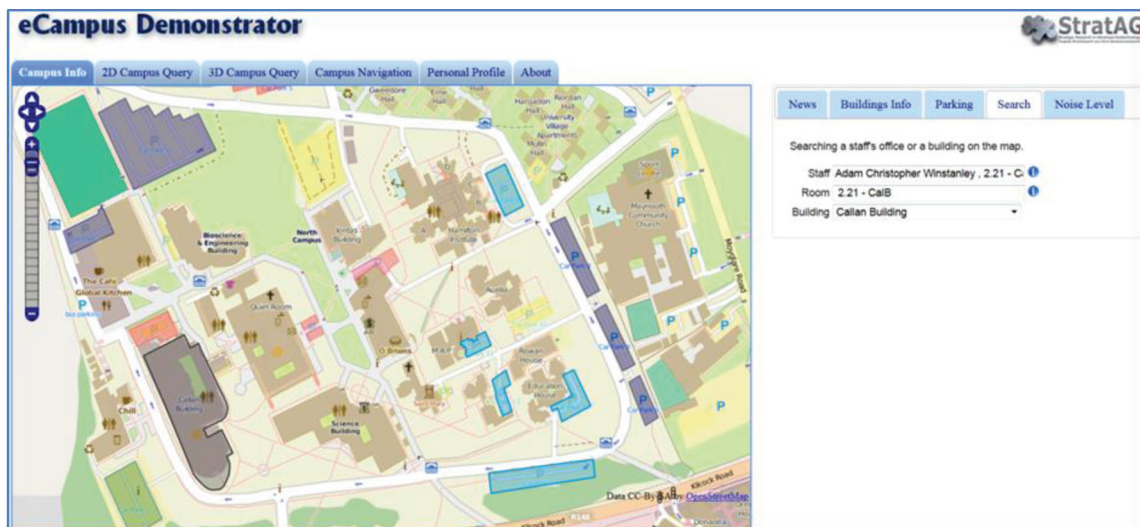


Figure 14. Staff search





## 2D Queries

2D Query functionality allows to perform Isovist, Field-of-View and Point-to-Select queries. User choose a query type and then click on a location on the map to indicate their view location. Depending on the query option, user may enter their field-of-view angle. The result of the query is displayed as blue bounded buildings and the field-of-view shaded in red. Users can click on the blue bounded buildings to see a description of the building's details.

A 2D Isovist is a 360 degree view around a users location out to a defined distance. For example, users can send an Isovist query like “Retrieve all objects around me that I can actually within 50 meters”. Users ask this by clicking a location on the map and dragging the mouse out to a desired distance. The query is then sent and the results returned as all buildings visible get highlighted on the map (Figure 16).

Users can explore further information of the resulting buildings by clicking on them similar to the main Campus exploring function (Figure 17).

By constricting the 360° IsoVist query, the Field-of-View query allows users to indicate the viewshed angle, the direction of their query, and a distance from their chosen location. The application will retrieve all objects visible within that angular Field-of-View and direction as well as that distance (Figure 18).

The Point-to-Select query retrieves objects intersecting a “line-of-sight” defined by clicking 2 locations on the map. User can also define a zigzag line containing multiple locations to assume a path from one location to another (Figure 19).

The mobile device interface for 2D queries takes into account the current user location (from GPS), tilt of the device (from accelerometer) and azimuth of pointing direction (from digital compass) (Figure 20).

Figure 15. Campus exploration on SmartPhone

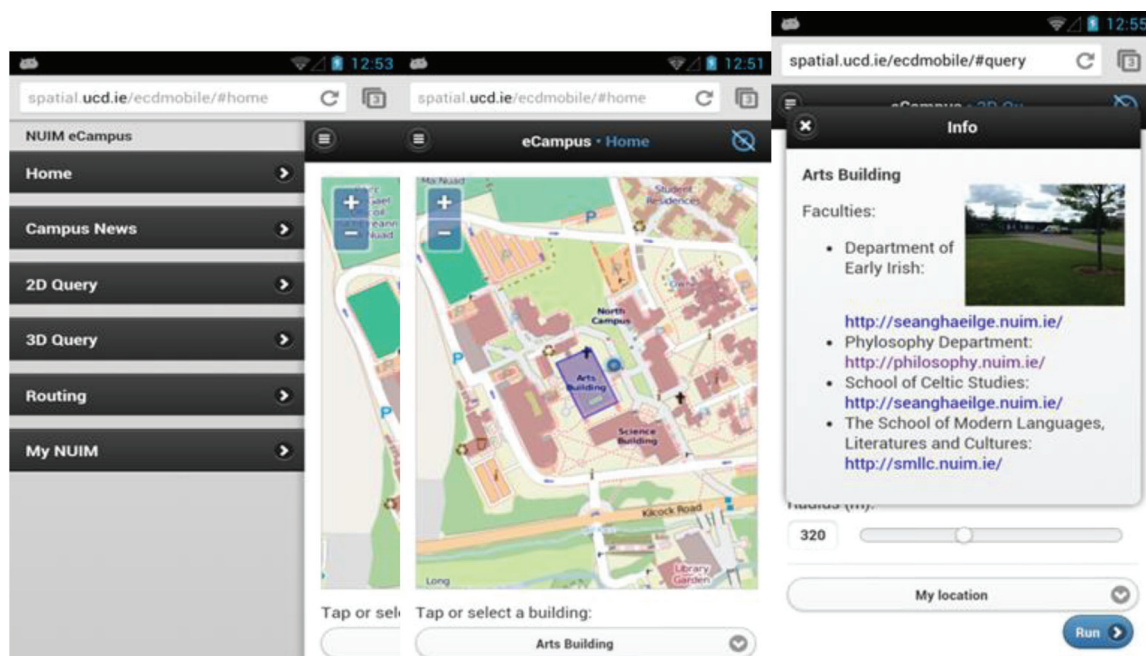


Figure 16. Isovist view query

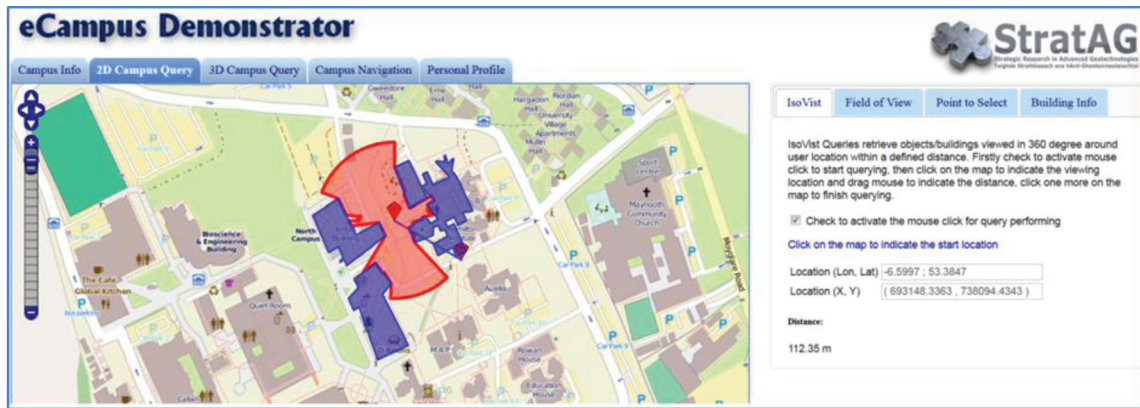


Figure 17. Building information after querying

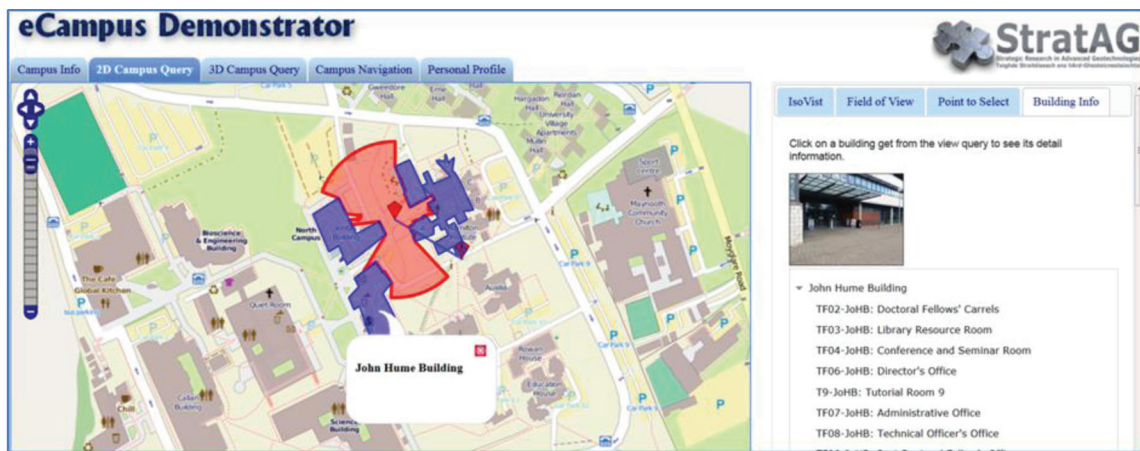
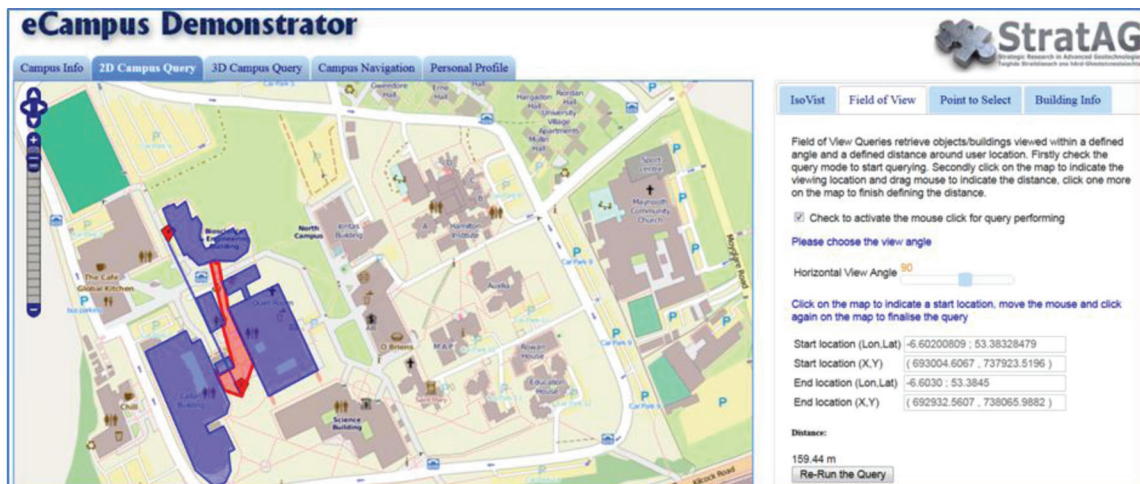


Figure 18. Field-of-View query



## A Case Study for eCampus Spatial

Figure 19. 2D Point-to-Select query

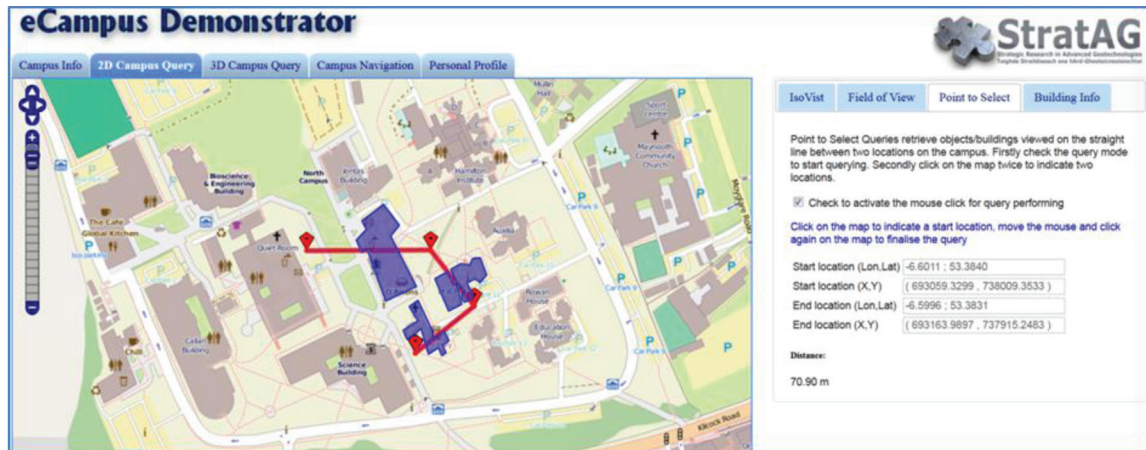
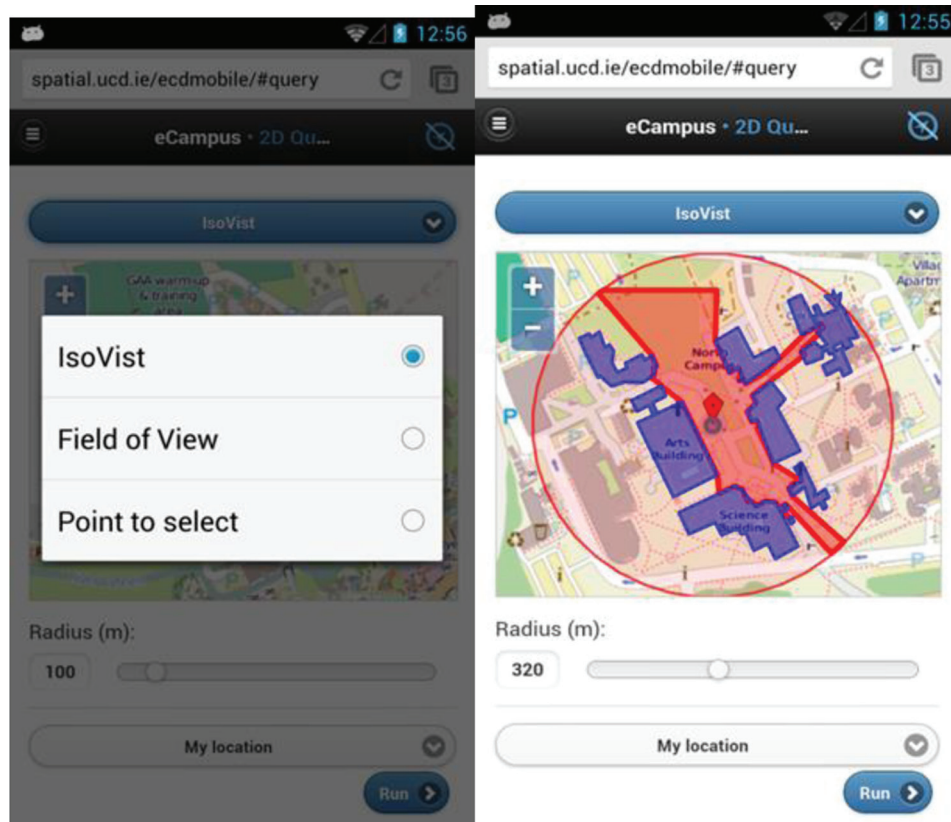


Figure 20. 2D Isovist query on a mobile device





### 3D Queries

The 3D Query functionality displays the NUIM campus in 3D using Google Earth (3D map). At this stage, users can query the campus by clicking directly on a building or a window (i.e. room). The corresponding building information or room information will be displayed (Figure 21).

The eCampus application provides two 3D query types: 360° Isovist view (Threat Dome) and Frustum view. The 360° Isovist view is a 3D version of the 2D Isovist view, which means it returns objects that users can actually see around them in all directions horizontal and vertical out to a specified distance (Figure 22).

Figure 21. Campus exploration in 3D

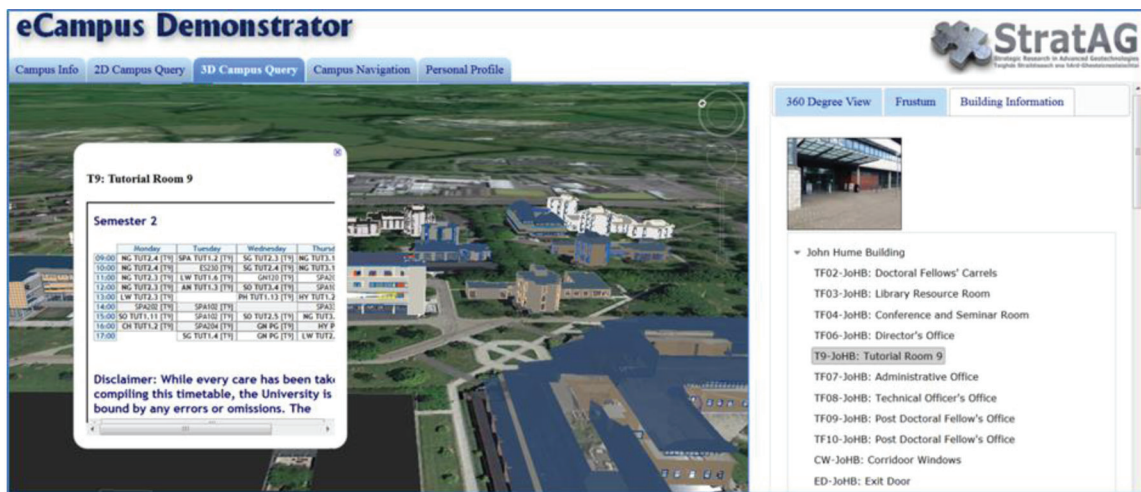
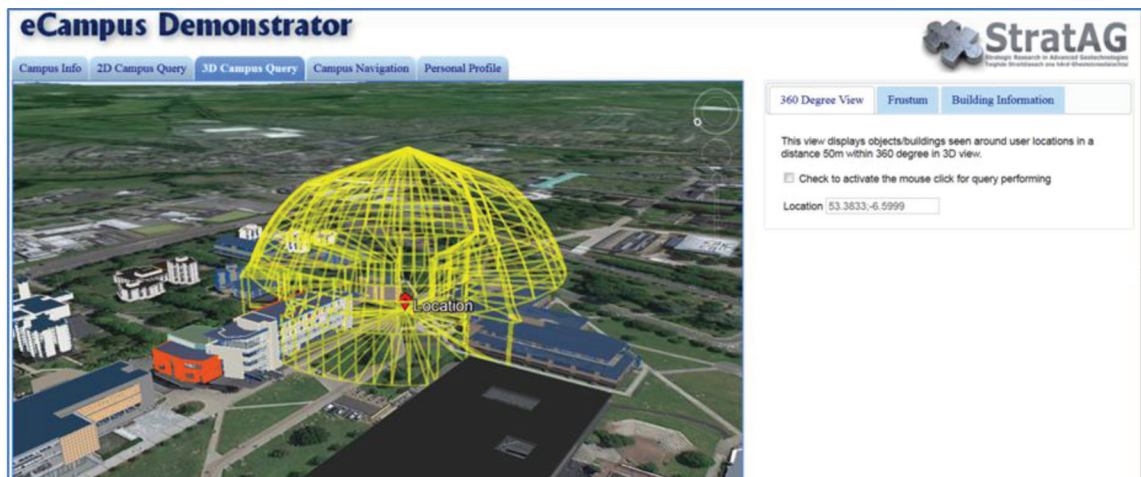


Figure 22. 3D 360° Isovist view query



## A Case Study for eCampus Spatial

The Frustum view query is the 3D version of 2D Field-of-View. It constrains the visible query space to a location having vertical and horizontal angular visibility and tilt in a particular direction and out to a specified distance. The intersection of the visible frustum query space with buildings and other database objects represent what users can actually see (Figure 23).

For the mobile version, the 360° Isovist and Frustum query interfaces look the same as in 2D, but the query results provide a list of buildings and rooms intersected by the view (Figure 24) (i.e., the parts intersected by the yellow Threat Dome (Figure 22) or by the green pyramid (Figure 23) and the building(s)).

## Path Navigation

This functionality helps users find a route between two locations on the campus, it is available for both 2D and 3D maps. A location may be a building name, a room name or any location defined by longitude and latitude coordinates from a mouse click (Figure 25). The system provides various navigation options that users can select on such as pedestrian route, driving route, wheelchair route or route with directional images.

Route navigation with directional images shows users an augmented (with arrows) pedestrian route of thumbnail images along the route. The thumbnail images show the image of locations nearby and the direction to follow (arrow) to reach their destination. (Figure 26).

Figure 27 illustrates a different option of navigating with directional images. Users can choose the start and end location by indicating a building name and/or a room number from a list.

Regarding the mobile version of route navigation, users can ask to find a route from their current location based on their GPS location to a building/room or between two buildings on the campus (Figure 28).

Figure 23. 3D Frustum query

**eCampus Demonstrator**

Campus Info 2D Campus Query 3D Campus Query Campus Navigation Personal Profile

**T5: Tutorial Room 5**

**T5: Tutorial Room 5**  
John Hume Building  
North Campus of National University of Ireland,  
Maynooth  
Ireland

Step 1 - Select Venue to T5 - Tutorial Room 5 (Change)  
Step 2 - Select Semester to Semester 1 (Change)

Please see degree structures for further information. All optional modules not be available to you. Note that practical sessions, small group work, modules are not identified. Contact relevant department for details.

• Advisory: This is the 2012/2013 Timetable. Students should check timetable daily. Semester 2 commences on Monday January 28th.

**Semester 1**

Monday	Tuesday	Wednesday	Thursday
09:00 FR TUFT.1 (1)	FR TUFT.11 (1)	POE 6 (1)	SD TUFT.3
10:00 MC TUFT.1 (1)	SPAZ22 (1)	POE 6 (1)	MY TUFT.3
11:00 FR TUFT.1 (1)	GC TUFT.12 (1)	POE 6 (1)	SPAZ20
12:00 SPAZ22 (1)	NO TUFT.14 (1)	POE 6 (1)	SPAZ20
13:00 MP TUFT.2 (1)	MY TUFT.12 (1)	MC TUFT.2 (1)	PA TUFT.3
14:00 MY TUFT.1 (1)	SPAZ21 (1)	FR TUFT.5 (1)	MP TUFT.3

**360 Degree View Frustum Building Information**

Frustum displays objects/buildings seen from user location according to user's defined direction (angle to north) and user's defined distance in 3D view.

☐ Check to activate the mouse click for query performing

Height Angle: 10  
Width Angle: 15  
Tilt: 10  
Direction: 27  
Distance: 53  
Location: 53.3835;-6.6006  
Re-Run the query

Figure 24. 3D Isovist query and the result. Note query result is shown overlaid in 2D OSM as GE does not yet allow functionality for customised 3D objects overlaid on their mobile 3D maps

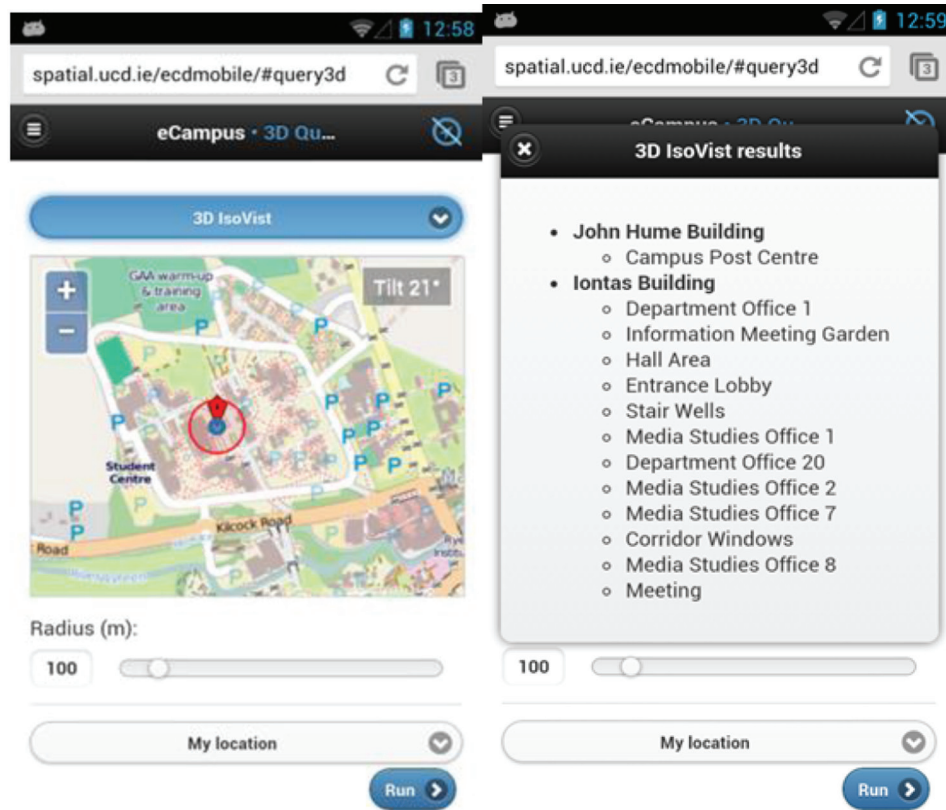
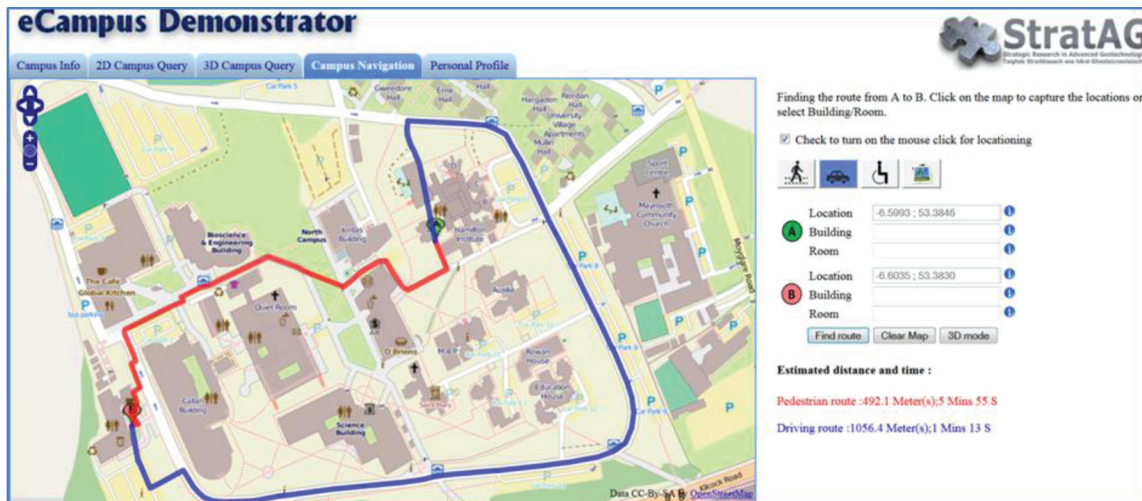


Figure 25. 2D Route navigation. Screen shot of a pedestrian (red) and driving (blue) route navigation between two locations from clicking on the map





## A Case Study for eCampus Spatial

Figure 26. Navigation with visualisation



Figure 27. 2D Route navigation with directional images between selected buildings/rooms



## 3D Route Navigation

Due to custom object display limitations within mobile GE, 3D route navigation is specifically for the desktop version of the Demonstrator. It is the same as for 2D map, except that the campus buildings are displayed in 3D (Figure 29).

## Personalisation

This functionality provides information specifically targeted to user interests. After logging into the system with student/staff ID, personal information such as a personal course calendar and news/events are personalized based on their chosen interests (e.g. History, Sports, Restaurants, etc.) (Figure 30).

Figure 28. Directional Image Routing on mobile device

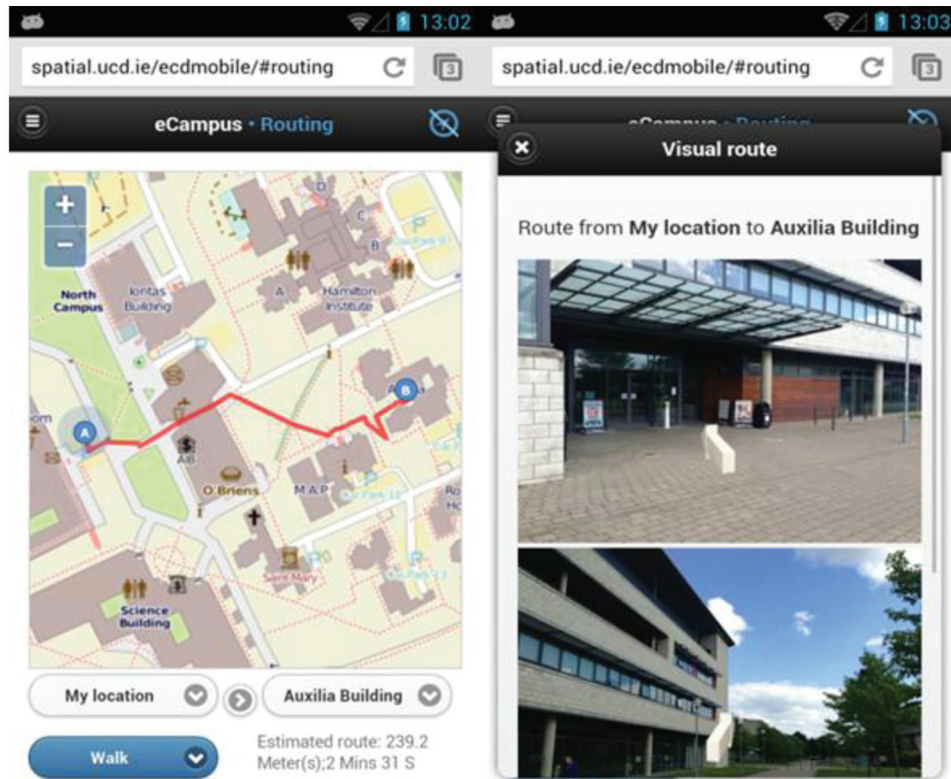
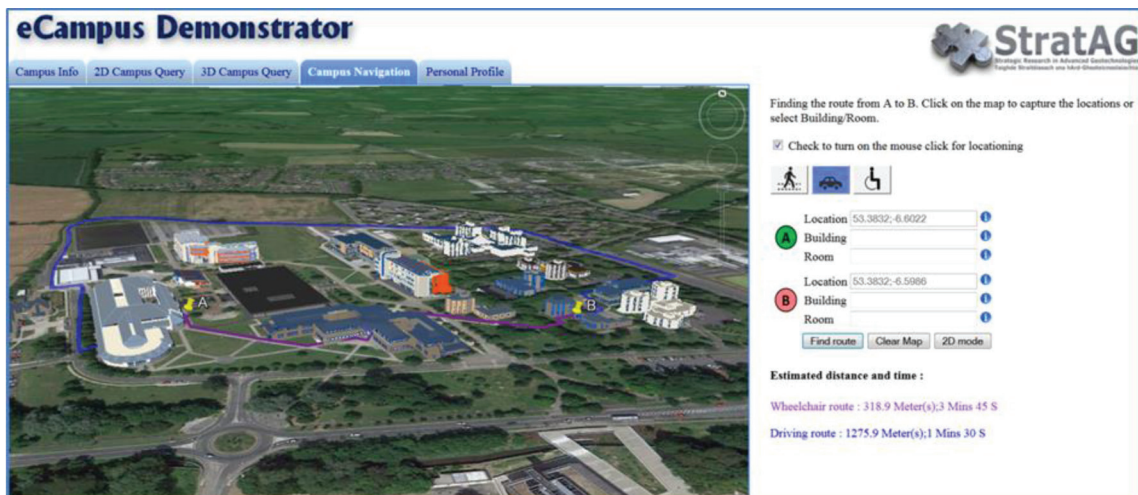


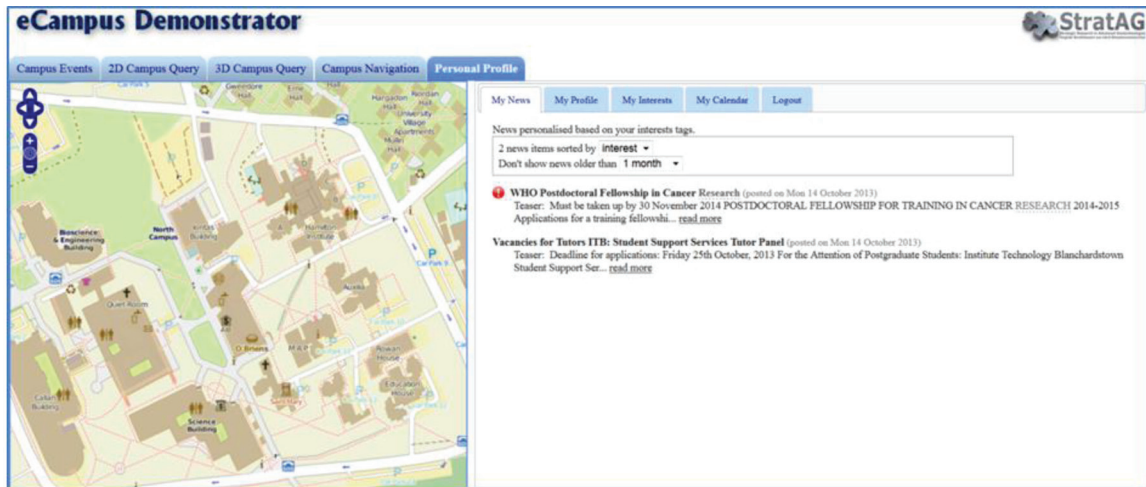
Figure 29. Route navigation in 3D





## A Case Study for eCampus Spatial

Figure 30. Personalised news

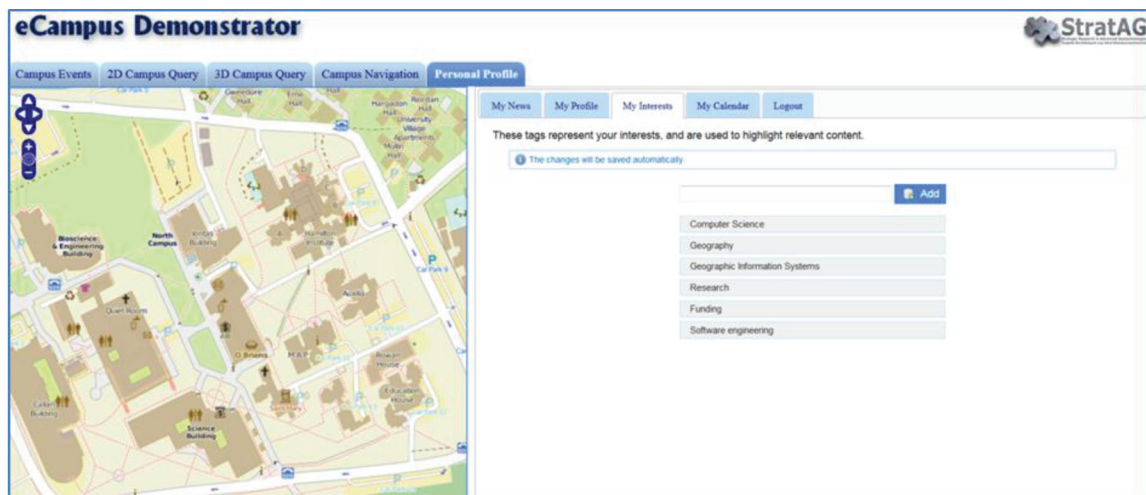


Users manage their interests by adding or removing them from their profile. An auto complete list of keywords is provided to help users in selecting and adding their interest keywords (Figure 31).

## DISCUSSION

During the development phase of this application, some important limiting factors emerged regarding the technology employed. After first analyzing the capabilities of the chosen technologies for implementation, the following discussion presents some advantages and disadvantages of our approach.

Figure 31. Personal interests



### 3D BI Modelling and Visualisation

When creating 3D virtual cities for Internet of Things applications and general mobile spatial interaction, the most important task is to generate detailed and geometrically accurate building models. In contrast to traditional methods (e.g. on-site surveying), TLS is an attractive alternative for collecting building coordinate data in terms of field time and accuracy (Truong-Hong, 2011), (Truong-Hong et al., 2012a), (Truong-Hong et al., 2012b). However, the process of building detailed 3D models from this point cloud data is still quite a manual process. In our application, it takes around 6-8 hours for each building. This implies automatic or semi-automatic processes must be developed to reconstruct building models with LoD3 to reduce post-processing bottlenecks for city-wide 3D modeling workflows used in this type of application.

As the GE platform currently only displays 3D objects above the earth's surface, this restricts visualisation of any underground detail below ground elevation level. To display and interact with entire building models above and below ground, an alternate web mapping platform should be considered.

At the present, mobile device browsers do not support the Google Earth API, therefore there is a limitation for visualisation of 3D models and other customised vector objects on mobile devices.

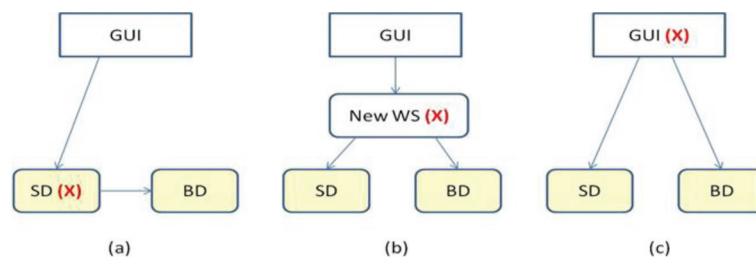
### Integration Level of Spatial Data and Business Data

As mentioned previously, the integration of spatial data and business data is performed at the client side, i.e. at the visualisation level. We considered three options to provide spatial data and related business data to users as shown in Figure 32.

In Figure 32:

- A. **Early-Integration:** In this approach, retrieving business data is performed from inside the spatial data retrieving WS. The final results sent back to GUI include spatial data and business data.
- B. **Aggregated Web-Service:** A new web-service is developed to compose the results returned by the spatial data web-service and the business data web- service.
- C. **Integration at the Visualisation Level:** The results of the spatial data web-service and business data web-service are overlaid at the visualisation level. We have chosen the third approach: integration at the visualisation level, as this approach provides some advantages compared to the other two approaches (Table 2).

Figure 32. Different spatial data and business data integration approaches. The cross (X) represents the integration point. The arrow describes the calling direction. WS: Web-Service; SD: spatial data WS; BD: business data WS; GUI: Graphical User Interface



*Table 2. Spatial data and business data integration options*

(a) Early-integration	There is a dependency of SD retrieving WS and a specific BD retrieving WS. That means SD WS cannot be reused for other purposes
(b) Aggregated web-service	SD and BD WS are independent. It depends on the user needs that the aggregated WS (AggWS) will integrate suitably SD and BD WS. Suppose $fA(WS)$ is the cost of analyzing the result of a WS, $fC(WS)$ is the cost of calling a web-service, then the cost of this approach to display the result is: $fC(AggWS) + fA(AggWS) + fC(SD) + fA(SD) + fC(BD)$ Note that the results of AggWS needs to be analyzed to draw the geometry shapes and the business data is then added to the feature data of the geometry object. In case the results of AggWS is in KML format, there is no need of analysing AggWS (so no cost), all spatial data and business data can be visualized. However, in that case the visualisation is fixed according to the API provided.
(c) Integration at visualisation level	SD and BD WS are independent. It depends on user needs that suitable SD and BD WS are consumed at the visualisation (client side code source). The cost to display the result to the users is: $fC(SD) + fA(SD) + fC(BD)$ Visualisation of the results is flexible according to the users' needs.

## OGC Services

The Open Geospatial Consortium (OGC) has developed some standards for geospatial processing technologies to enable applications from different commercial vendors to interoperate. However, the locationing services developed within OGC focus mainly on tracking and location-based applications for mobile devices (OGC, 2007). These services are far from what we require in this application, where all *RESTful* web-services for location-dependent directional and Vista space querying have been developed in our own algorithms.

## ROA Instead of SOA

For the last decade, Service Oriented Architectures (SOA) has been widely used for distributed applications, particularly on the Web. In Geographic Information Systems (GIS), there is no exception here. For instance in (Alameh, 2003), a GIS web-service architecture was proposed based on SOA technology. However, while SOA is a proven approach, in some cases it can be overly complicated thus processor heavy. For example, when handling a SOAP message, the client (desktop or mobile) needs to send a request with parameters constructed and wrapped in XML format with special headers and other elements. It also has to parse any response from the server in the same effusive XML format (Snell, Tidwell and Kulchenko, 2001). But in the case where the client is a mobile device, this approach contains far too much processing overhead in terms of the volume of data, most of it quite unnecessary, that must be sent/received on mobile devices having relatively limited wireless connection speeds and often a data transmission cost (Yin and Carswell, 2011). In this respect, JSON is a much lighter data format in terms of processing and transmitting wirelessly. Furthermore, we agree with the general statement that the *REST* architecture provides a “scalable and simple deployment of web-services and particularly appealing for Earth and Space Science” (Mazzetti et al., 2009) as *RESTful* web-services have been much used in geo-information sharing.

## **Dependency of 3D Query Performance and 3D Data Details**

In our application, users carry out spatial queries from outside buildings. Therefore only the geometries of exterior structural components of the building (e.g. facades, roofs, windows, doors, balcony, canopy and so on) associated with room level attribution (e.g. room name and function) were loaded into the database. In this way, it helps to reduce the complexity of the 3D models and thus improve 3D spatial query performance.

## **Other Limitations**

In theory, the HTTP protocol does not limit the length of a URI (according to RFC 2616 - Hypertext Transfer Protocol - HTTP/1.1). However some browsers may limit extremely long URI text strings, therefore the application should take into account this issue when designing *RESTful* web-services.

We also need more testing with student/staff users to get their feedback on the functionality as well as the performance of our application. Semantic Web technologies might also be employed to facilitate the integration of heterogeneous data (Ballatore, Wilson and Bertolotto, 2013).

## **CONCLUSION**

Providing users with business context data in location dependent queries helps to fulfil more users needs within detailed data environments. In order to obtain that objective, there is a need of 3D building modelling to at least LoD3. The workflow proposed in this chapter was successful in reconstructing geometrically accurate building models with LoD3. However, the procedure is time consuming for larger project areas where numerous building models need to be reconstructed; therefore, automation of this approach is still an open problem.

The flexibility, interoperability and heterogeneity of this kind of linked search application demands a suitable software architecture. In particular to this geospatial application, a Resource Oriented Architecture (ROA) was chosen for the implementation.

To the best of our knowledge, the eCampus Demonstrator presented in this chapter is among the first applications to allow users to explore in detail an area in both 2D and 3D maps, providing users with more utilities and directional search functionality than existing systems. However, there are still some limitations needed to be solved in the future as discussed.

This work can be considered as a starting point for developers and researchers when developing for similar application domains, such as when exploring a business park, hospital, or a shopping centre, etc.

## **ACKNOWLEDGMENT**

Research presented in this paper was funded by a Strategic Research Cluster Grant(07/SRC/I1168) by Science Foundation Ireland under the National Development Plan. The authors gratefully acknowledge this support.

## REFERENCES

- Alameh, N. (2003). Chaining Geographic Information Web-services. *IEEE Internet Computing*, 7(5), 22–29. doi:10.1109/MIC.2003.1232514
- Ballatore, A., Wilson, D. C., & Bertolotto, M. A. (2013). Survey of Volunteered Open Geo-Knowledge Bases in the Semantic Web. *Quality Issues in the Management of Web Information*, 50, 93–120.
- Carswell, J., Gardiner, K., & Yin, J. (2010). Mobile Visibility Querying for LBS. *Transactions in GIS Journal*, 14(6), 791–809. doi:10.1111/j.1467-9671.2010.01230.x
- Carswell, J. D. (2010). 3DQ: Threat Dome Visibility Querying on Mobile Devices. *GIM International*, 24(8), 24.
- Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures* (PhD dissertation). University of California, Irvine, CA.
- Fielding, R. T., & Taylor, R. N. (2002). Principal Design of Modern Web architecture. *ACM Transactions on Internet Technology*, 2(2), 115–150. doi:10.1145/514183.514185
- FME. (2013). *Safe software: FME desktop*. FME.
- Guinard, D., Trifa, V., & Wilde, E. (2010). A Resource Oriented Architecture for the Web of things. *IEEE International Conference on Internet of Things*. doi:10.1109/IOT.2010.5678452
- Haala, N., & Kada, M. (2010). An Update on Automatic 3D Building Reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(6), 570–580. doi:10.1016/j.isprsjprs.2010.09.006
- Henn, A., Gröger, G., Stroh, V., & Plümer, L. (2013). Model driven reconstruction of roofs from sparse LIDAR point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 76, 17–29. doi:10.1016/j.isprsjprs.2012.11.004
- Isikdag & Zlatanova. (2010). *Interactive modelling of buildings in Google Earth: A 3D tool for Urban Planning*. In *Developments in 3D Geo-Information Sciences* (pp. 52–70). Springer Berlin Heidelberg.
- Jacob, R., Zheng, J., Ciepluch, B., Mooney, P., & Winstanley, A. C. (2009). Campus Guidance System for International Conferences Based on OpenStreetMap. *Proceeding W2GIS '09*, 187–198. doi:10.1007/978-3-642-10601-9\_13
- Kaharaman, I., Karas, I. R., & Rahman, A. A. (2011). *Developing web-based 3D Campus Information System*. ISG & ISPRS.
- Kent State University Campus Maps. (n.d.). Retrieved from <http://www.kent.edu/campuses/maps/map.cfm>
- Kurtagic, H., Birch, J., & Zeiss, G. (2009). *An Open Architecture for RESTful Geospatial Web-services*. FOSS4G, Sydney, Australia.
- Leica Geosystems, A. G. (2011). *Leica Geosystems AG: Leica Cyclone-3D Point Cloud Processing Software Leica Geosystems AG*. Leica CloudWorx for AutoCAD.

- Lucchi, R., Millot, M., & Elfers, C. (2008). *Resource Oriented Architecture and REST*. JRC Scientific and Technical Report.
- Mazzetti, P., Nativi, S., & Caron, J. (2009). RESTful Implementation of geospatial services for Earth and Space Science applications. *International Journal of Digital Earth*, 2(sup1Supplement 1), 40–61. doi:10.1080/17538940902866153
- Musliman, I. A., Abdul-Rahman, A., & Coors, V. (2010). Incorporating 3D-GIS Spatial Operator with Building Information Models in Construction Management using Geo-DBMS. *5th International Conference on 3D GeoInformation*, 38, 147-154.
- OGC. (2007). *OGC Report- Summary of OGC Web-services, Phase 4, Interoperability Testbed*. Retrieved from <http://www.opengeospatial.org/projects/initiatives/ows-4/>
- Percivall, G. (Ed.). (2002). *OpenGIS Abstract Specification Topic 12: OpenGIS Service Architecture*. Retrieved May 2013 from <http://www.opengeospatial.org/standards/as>
- Pham Thi, T. T., Truong-Hong, L., Yin, J., & Carswell, J. (2013). Exploring Spatial Business Data: A ROA Based eCampus Application. *LNCS*, 7820, 164–179.
- Snell, J., Tidwell, D., & Kulchenko, P. (2001). *Programming Web-services with SOAP*. O'Reilly Publisher.
- Truong-Hong, L. (2011). Automatic Generation of Solid Models of Building Façades from LiDAR Data for Computational Modelling. School of Architecture, Landscape and Civil Engineering, University College Dublin.
- Truong-Hong, L., Pham Thi, T. T., Yin, J., & Carswell, J. (2013). Detailed 3D Building Models for Google Earth Integration. *ICCSA*. Springer-Verlag.
- Truong-Hong, L., Laefer, D. F., Hinks, T., & Carr, H. (2012). Flying Voxel Method with Delaunay Triangulation Criterion for Façade/Feature Detection for Computation. *Journal of Computing in Civil Engineering*, 26(6), 691–707. doi:10.1061/(ASCE)CP.1943-5487.0000188
- Truong-Hong, L., Laefer, D. F., Hinks, T., & Carr, H. (2012). Combining an Angle Criterion with Voxelization and the Flying Voxel Method in Reconstructing Building Models from LiDAR Data. *Computer-Aided Civil and Infrastructure Engineering*. doi:10.1111/j.1467-8667.2012.00761.x
- University College Dublin Mobile Services. (n.d.). Retrieved from <http://www.ucd.ie/itservices/itsupport/mobileservices>
- Yin, J., & Carswell, J. D. (2011). Touch2Query enabled mobile devices: a case study using OpenStreet-Map and iPhone. In *Web & Wireless Geographical Information System* (pp. 203–218). Kyoto, Japan: Springer. doi:10.1007/978-3-642-19173-2\_16
- Zlatanova, S. (2006). 3D geometries in spatial DBMS. 3D-GIS.